PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-251379

(43)Date of publication of application : 22.09.1997

(51)Int.Cl. G06F 9/06
G06F 3/14
G06F 9/46

(21)Application number : 08-061464

(22)Date of filing : 18.03.1996

(71)Applicant : TOSHIBA CORP
TOSHIBA SOFTWARE ENG KK

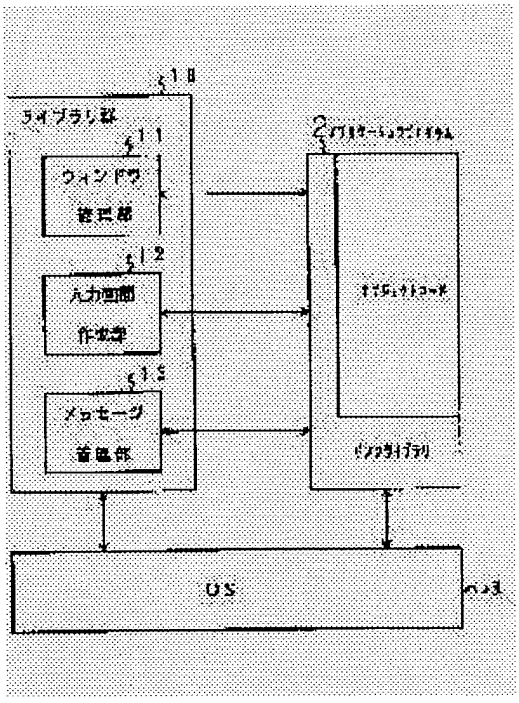
(72)Inventor : SHIMIZU NOBUO
KONDO YUJI

(54) COMPUTER SYSTEM

(57)Abstract:

PROBLEM TO BE SOLVED: To easily transplant an application program not coping with a window system to the window system.

SOLUTION: The application program 2 not coping with the window system is linked with a library group 10 for transplanting and a window management part 11 prepares a window when the application program 2 is activated. Also, when the application program 2 requests the display of an input screen, an input screen preparation part 12 plots the input screen inside the window and a message management part 13 activates a message loop for receiving a message from an OS 3. Then, when data input from a user is present, by delivering the input data to the application program 2 and ending the message loop, control is returned to the application program 2.



LEGAL STATUS

[Date of request for examination] 13.03.2003

[Date of sending the examiner's decision of

rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

* NOTICES *

1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DETAILED DESCRIPTION

[Detailed Description of the Invention]
[0001]
[Field of the Invention] This invention relates to the computer system which comes to apply the operating system which executes an application program separate in a break and each window for a display in two or more windows, and relates to the computer system which makes it possible to transplant easily to a window system the application program corresponding to un-with which the correspondence for operating especially in this window is not made.
[0002]
[Description of the Prior Art] In recent years, various personal computers, such as laptop including a desktop, a notebook, and a palm top, are developed, and the spread has a remarkable thing. And various devices have been made by the interface of a computer system and a user with these spread.
[0003] In the former, the interface called CUI (Character User Interface) etc. is in use, and the input screen the application program under activation urges an entry of data to a user by the system was displayed, and when a user inputted KOMANTO through a keyboard by one side, it was said that an interface was secured.
[0004] However, according to such an approach, there is a problem that a command for the user itself to have a dialog with an application program must be kept in mind for example, etc., and user-friendliness was seldom able to say that it was good.
[0005] And the interface called GUI (GraphicalUser Interface) etc. has appeared as what solves such a problem. This GUI displays a picture, an icon, etc. on a screen, enables it to operate them intelligibly intuitively, with pointing devices, such as a mouse, by specifying the displayed icon, can advance actuation and, recently, is becoming in use [this GUI / the interface of a computer system and a user].
[0006] A window system divides a scope into some apertures (window), and means the system which can operate separate application in each window. A screen is divided vertically and horizontally, what is assigned to each application program by using that each as a window, and the lap of a window are permitted to this window, two or more windows are created in it, and the thing of assigning each application program to each window of these plurality etc. exists in it.
[0007] On the other hand, in order to operate an application program by OS subordinate who realizes this window system, window correspondence must be made for the application program itself.
[0008] In a window system, as mentioned above, GUI is adopted, and a user performs almost all actuation with pointing devices, such as a mouse, by this. And in this window system, a pointing device can perform migration, size change, etc. of a window now, and OS serves as structure which also notifies such information to a working application program as a message in that window including the input data.
[0009] That is, the application program which operates under such an environment had to be equipped with the message reception function which acquires the message from OS, and had to be equipped with the message-processing function corresponding to the message of a parenthesis.
[0010] However, still, and in order to transplant to the bottom of the environment of a window C stem

which mentioned such an application program above, the user who utilizes as a resource the application program which does not support a window system needed correction on source program level, and when it was the application program of object offer, he had the problem of not being transplantable, by the end user.

[0011]

[Problem(s) to be Solved by the Invention] Thus, although the thing on condition of a window system is most, OS applied to the latest personal computer There are still many users who utilize as a resource the application program which does not support such a window system. moreover, in order to transplant to the bottom of the environment of a window USHII stem which mentioned such an application program above Correction on source program level was required, and when it was the application program of object offer, there was a problem of not being transplantable, in an end user.

[0012] It aims at offering the computer system which enables this invention to transplant easily to a window system the application program corresponding to un-with which it is made in view of such the actual condition, and the correspondence for operating by the window system is not made.

[0013]

[Means for Solving the Problem] This invention displays at least one or more windows on the same screen. It is the operating system which can execute a separate application program in each window. In the computer system which comes to apply the operating system which has the function to transmit various messages to a working application program in said window It is connected with the application program corresponding to un-with which the correspondence for operating in said window including the reception function of said message is not made. When said application program corresponding to un- requires the display of an input screen as a windowing means to require creation of a window of said operating system at the time of starting of said application program corresponding to un- An input-screen creation means to draw the input screen in the window created by said windowing means, It synchronizes with said input screen into said window by said input-screen creation means. A message loop starting means to start the message loop for receiving the message from said operating system, When the message which received with said message loop starting means is a thing containing the input data corresponding to the input screen in which said application program corresponding to un-carried out the display demand A message-processing means to terminate said message loop while transmitting the input data to said application program corresponding to un-, The library group which comes to have a window release means to require release of a window of said operating system at the time of termination of said application program corresponding to un-is provided. It is characterized by supposing that it is implantable under said window environment only by connecting the application program corresponding to un-with which said window correspondence is not made with said library group.

[0014] In case the application program corresponding to un-with which the correspondence for operating in a window is not made is transplanted to the bottom of a window system environment according to this invention, a user links that application program and library group for transplantation of this invention.

[0015] For example, when an application program uses DLL (Dynamic LinkLibrary), in order to determine which DLL the application program uses, an import library is linked to the creation time of an application program. And OS can recognize whether it is that to which that application program operates using which DLL by the link of this import library. Therefore, the application program corresponding to un-will be connected with the library group (DLL) for transplantation by the link of this import library in this case.

[0016] In the conventional application program, an input screen is displayed on a screen, the information which the user inputted to this screen is incorporated as input data, and successive actuation of advancing processing based on this incorporated input data is performed.

[0017] So, in this invention, the library group for transplantation transplants the application program corresponding to un-to a window system by substituting for windowing of the following [the following timing].

(1) It is the demand of creation of the window to OS at the time of starting.
(2) It is drawing of the input screen in a window at the time of the display of an input screen.
[0018] At the time of the message reception from the starting (3) OS of the message loop for receiving the message from OS
a) In the case of input data, it is a transfer of the input data to an application program.
[0019] Termination of a message loop (thereby, control is returned to an application program).
b) Windowing corresponding to the message the case of window control, such as migration.
(4) It is the demand of release of the window to OS at the time of termination.
[0020] Thereby, from OS, it is recognized as an application program of window system correspondence, and an application program can be processed incorporating input data with an old interface, and does not need any correction of an application program etc. further.
[0021]
[Embodiment of the Invention] Hereafter, 1 operation gestalt of this invention is explained with reference to a drawing. Drawing 1 is the block diagram showing the outline configuration of the computer system of this operation gestalt.
[0022] As for the computer system of this operation gestalt, the whole is controlled by OS (operating system)3 to be shown in drawing 1 . Moreover, the application program 2 which operates by this OS3 subordinate is related with the functional object needed among two or more functional objects (library groups 10a-10b) stored in DLL (Dynamic LinkLibrary)1, and operates. And the library group for transplantation by which it is characterized [of this operation gestalt] will also be stored in this DLL1, and the application program corresponding to un-[window system] will be related with the library group for this transplantation.
[0023] In addition, relating with these application programs 2 and the library group meant in DLL1 is realized by linking an import library (link library) to the creation time of an application program 2. and OS3 -- the link of this import library -- an application program 2 -- it recognizes whether it is what operates using which library ****, respectively.
[0024] The outline configuration of the library group for transplantation of this operation gestalt is shown in drawing 2 . As shown in drawing 2 , the library group 10 for transplantation of this operation gestalt possesses the window Management Department 11, the input-screen creation section 12, and the message management section 13, and becomes.
[0025] The window Management Department 11 demands release of a window from OS3 at the time of termination of the application program 2 corresponding to un-while demanding creation of a window from OS3 at the time of operation of the application program 2 corresponding to un-.
[0026] The input-screen creation section 12 draws the input screen in the window created by the window Management Department 11, when the application program 2 corresponding to un-requires the display of an input screen.
[0027] And the message management section 13 performs the turnover to the application program 2 corresponding to un-of input data, and closing of a message loop while it starts the message loop for receiving the message from OS3, for example, performs processing according to the classification of a message called migration, size change, etc. of a window, when the input-screen creation section 12 draws an input screen based on the display demand of the input screen by the application program 2 corresponding to un-.
[0028] Here, the operations sequence in the conventional environment of an application program where the correspondence to a window system is not made with reference to drawing 3 is explained. First, the application program corresponding to un-requires creation and a display of an input screen from OS (step F1). If a user performs the data input corresponding to this input screen, that data will be handed over from OS by the application program corresponding to un-, and the application program corresponding to un-will perform processing based on this handed-over input data (step F2).
[0029] That is, the application program corresponding to un-repeats step F1 - step F2 by required data, and advances successive processing. The principle of operation in the conventional environment of an application program where the correspondence to a window system is not made is shown in drawing 4 .

[0030] If the data which the user inputted from OS are transmitted after the application program corresponding to un-requires creation and a display of an input screen from OS ((1) of drawing 4) ((2) of drawing 4), the application program corresponding to un-will perform processing based on this transmitted input data ((3) of drawing 4).
[0031] Similarly, when the following data are required, creation and a display of an input screen are required from OS ((4) of drawing 4), and processing is performed based on the input data ((5) of drawing 4) returned to this demand.
[0032] Thus, it advances successive processing, the conventional application program corresponding to un-receiving required data. Next, the operations sequence of the application program with which the correspondence to a window system was made with reference to drawing 5 is explained.
[0033] First, a correspondence application program starts the message loop for inputting the message from OS (step G1). The reception of a message and the dispatch processing according to the classification of that message are included in this message loop.
[0034] After this message loop starting, when a correspondence application program stands by the message from OS (step G2) and a message is received, it judges whether it is the message (Y of step G3) and this message instruct termination to be (step G4), and if it is not the message which directs termination (N of step G4), processing based on the classification of that message will be performed (step G5).
[0035] Therefore, the module which performs that processing when the message about window control of migration of a window, size change, etc. is received at this time is dispatched, and when the message containing the input data from a user etc. is received, the module which processes that data is dispatched.
[0036] The principle of operation of the application program with which the correspondence to a window system was made is shown in drawing 6 . If a correspondence application program starts the message loop for inputting the message from OS ((1) of drawing 6), the data input from a user is begun after this, whenever events, such as migration, size change, etc. of a window, occur, a message will be transmitted from OS, and processing based on the message will be performed each time ((2) - (3) of drawing 6 , (4) - (5) and (6) -). And this processing is repeated until a correspondence application program is completed.
[0037] The operations sequence of the application program with which the correspondence to the window system in the environment (window system environment) of this operation gestalt is not made here is explained.
[0038] In the computer system of this operation gestalt, as shown in drawing 2 , an executive program is created by making the application program 2 corresponding to un-link with the library group 10 for transplantation. Hereafter, the operations sequence of this activation application is explained.
[0039] The operations sequence of the library group 10 for transplantation and the linked application program 2 corresponding to un-is shown in drawing 7 thru/or drawing 11 . Starting of the application program 2 corresponding to un-performs initial processing first (step A1 of drawing 7). at this time, with the application program 2 corresponding to un-, an indispensable function will be published at the time of starting [say / "Open" etc.], and issue of this function will be notified to the library group 10 for link attachment **** transplantation by control of OS3.
[0040] Actuation of the library group 10 for the transplantation when receiving this notice is shown in drawing 8 . By the library group 10 for transplantation which received this notice, the window Management Department 11 secures various resources for the application program 2 corresponding to un-to operate in a window system environment (step B1 of drawing 8). And the window Management Department 11 changes and performs creation of a window for the application program 2 corresponding to un-to operate to the application program 2 corresponding to un-(step B-2 of drawing 8).
[0041] then, if the function which requires a screen display for the application program 2 corresponding to un-to input data is published (step A2 of drawing 7), similarly, issue of this function will be notified to the library group 10 for link attachment **** transplantation by control of OS3, and the input-screen creation section 12 will perform actuation as shown in drawing 9 by the library group 10 for

transplantation which received this notice.
[0042] That is, the call of a window procedure is performed to OS3 (step C1 of drawing 9), and drawing processing within the window procedure based on the display demand received from the application program 2 corresponding to un-is performed (step C2 of drawing 9).
[0043] And if the entry of data through this displayed screen is required from the application program 2 corresponding to un-, by the library group 10 for transplantation, the message management section 13 will perform processing shown in drawing 10 .
[0044] First, the message management section 13 starts the message loop for receiving the message which transmits to the window where OS3 was created by the window Management Department 11 (step D1 of drawing 10). While becoming ability ready for receiving about the message containing the data which the user inputted by this corresponding to the display screen drawn by the input-screen creation section 12, the message which directs migration, size change, etc. of a window also serves as ability ready for receiving.
[0045] And the message management section 13 performs dispatch according to the received message, and performs the processing (step D2 of drawing 10). In addition, as a result of checking the key input by the window procedure, when it is input data from a user, the keycode inputted into the internal key buffer which can refer to the application program 2 corresponding to un-is registered (step D3 of drawing 10), and a message loop is terminated (step D4 of drawing 10). That is, control will be returned to the application program 2 side corresponding to un-by termination of this message loop, and the activation by the side of the application program 2 corresponding to un-will be resumed.
[0046] And when the application program 2 corresponding to un-performs a post process, in (step A4 of drawing 7), and the library group 10 for transplantation, the window Management Department 11 processes in a procedure as shown in drawing 11 .
[0047] The window Management Department 11 releases the various resources secured at the time of termination of the application program 2 corresponding to un-(step E1 of drawing 11), and demands destruction of a window from OS3 (step E2 of drawing 11).
[0048] By this, the correction with a source level etc. will be unnecessary at all, and will become, and the application program 2 corresponding to un-will be recognized as an application program of window system correspondence from OS3, as a result of being able to operate without being conscious of the environment of a window system.
[0049] The principle of operation at this time is explained with reference to drawing 12 . If an application program 2 is started and initialization processing is performed ((1) of drawing 12), the window Management Department 11 of the library group 10 for transplantation will demand creation of the window for application program 2 etc. from OS3 ((2) of drawing 12). If processing termination of the window Management Department 11 here is notified to an application program 2 ((3) of drawing 12) next, an application program 2 will notify the screen creation demand for data inputs to the library group for transplantation ((4) of drawing 12). By the library group 10 for transplantation which received this notice, the input-screen creation section 12 draws in the window which it judged [window] whether an alphabetic character and a ruled line should have been drawn in which location of the window created previously, and had that screen for data inputs created ((5) of drawing 12). At this time, the input-screen creation section 12 publishes the demand which writes an alphabetic character to OS3 per an alphabetic character unit or character string according to a situation. And the input-screen creation section 12 will return processing to an application program 2, if all the drawing demands are filled ((6) of drawing 12).
[0050] An application program 2 will publish a data input demand to the library group 10 for transplantation, if an entry of data is needed during the processing ((7) of drawing 12). By the library group 10 for transplantation which received this demand, the message management section 13 starts the message loop for receiving a message from OS3 ((8) of drawing 12).
[0051] In addition, by the window system, receiving the message which shows messages other than the demanded data input, for example, migration of a window, modification of size, etc., is also considered at this time ((9) of drawing 12). In this case, by the library group 10 for transplantation, processing

which fills the demand of this received message is performed ((10) of drawing 12), and the entry of data expected further is stood by.
[0052] Moreover, when the message containing that data to expect is received ((11) of drawing 12), the message management section 13 hands over the input data contained in this message to an application program 2 while terminating a message loop ((12) of drawing 12) ((13) of drawing 12). Thereby, control is returned to an application program 2.
[0053] And an application program 2 performs processing based on this received input data. When you need the data input from this or subsequent ones, for example, a user, it repeats the same processing again. And when an application program 2 performs a post process, the window Management Department 11 of the library group 10 for transplantation performs release of the various resources secured at the time of starting of an application program 2, and created destructive processing of a window.
[0054] In addition, in order to return control to the application program 2 corresponding to un-by terminating a message loop with this operation gestalt, For example, although reception of this message can carry out immediately when migration of a window is directed after returning control to the application program 2 corresponding to un- Next, since this windowing will be performed first of all when an entry-of-data demand is performed from the application program 2 corresponding to un-, it is thought that it is satisfactory.
[0055] By this, a user can transplant to a window system the application program which does not support a window system easily and cheaply.
[0056]
[Effect of the Invention] Since the application program corresponding to un-with which window correspondence is not made is transplantable to the bottom of windowing environment only by connecting with the library group for transplantation according to this invention as explained in full detail above, transplantation can be performed easily and cheaply, without also needing the knowledge for the correction with a source level etc. becoming entirely unnecessary, and making it correspond to a window system.

[Translation done.]

JAPANESE [JP,09-251379,A]

CLAIMS DETAILED DESCRIPTION TECHNICAL FIELD PRIOR ART EFFECT OF THE INVENTION TECHNICAL PROBLEM MEANS DESCRIPTION OF DRAWINGS DRAWINGS

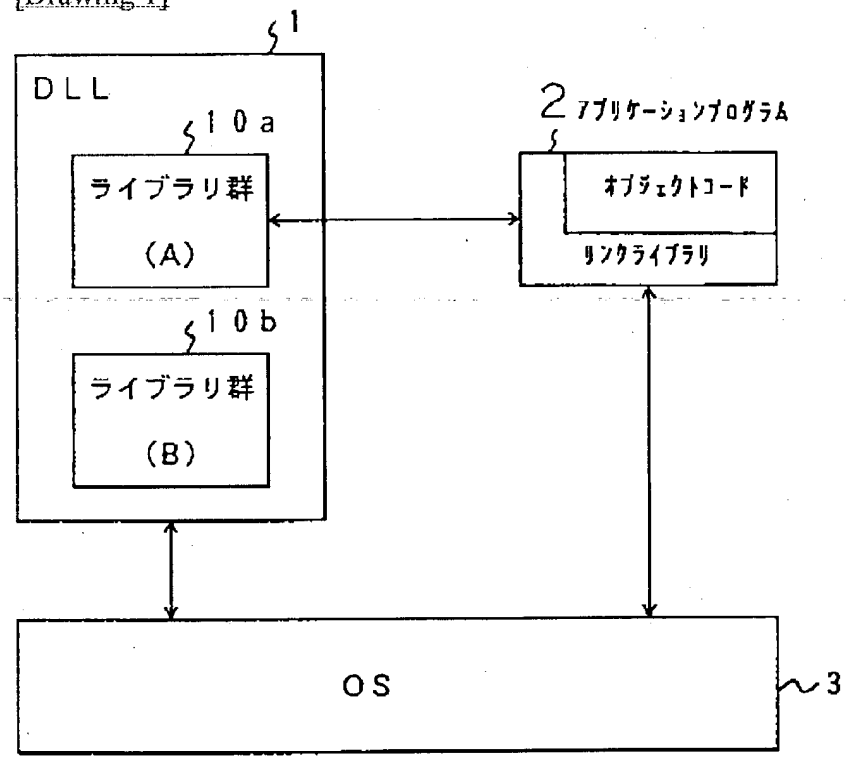
[Translation done.]

* NOTICES *

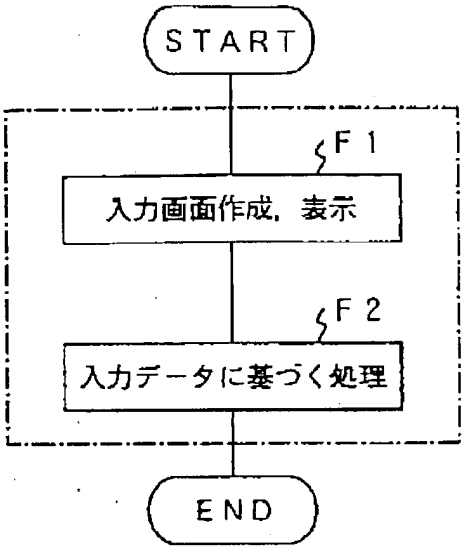
1.This document has been translated by computer. So the translation may not reflect the original precisely.
2.**** shows the word which can not be translated.
3.In the drawings, any words are not translated.

DRAWINGS

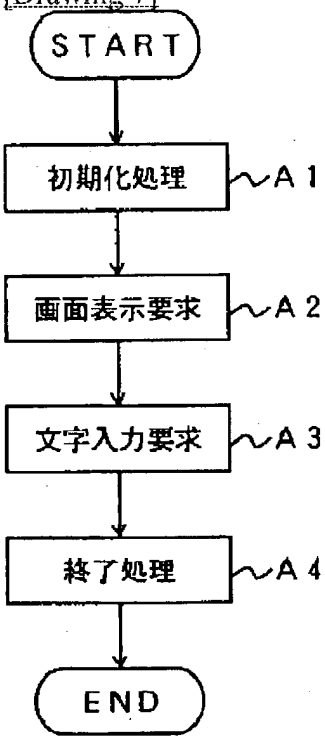
[Drawing 1]



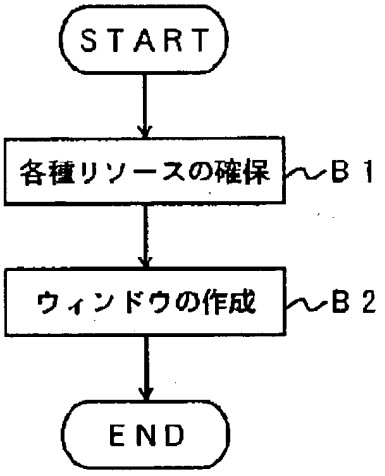
[Drawing 3]



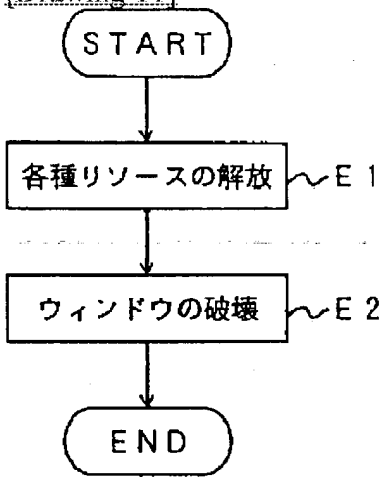
[Drawing 7]



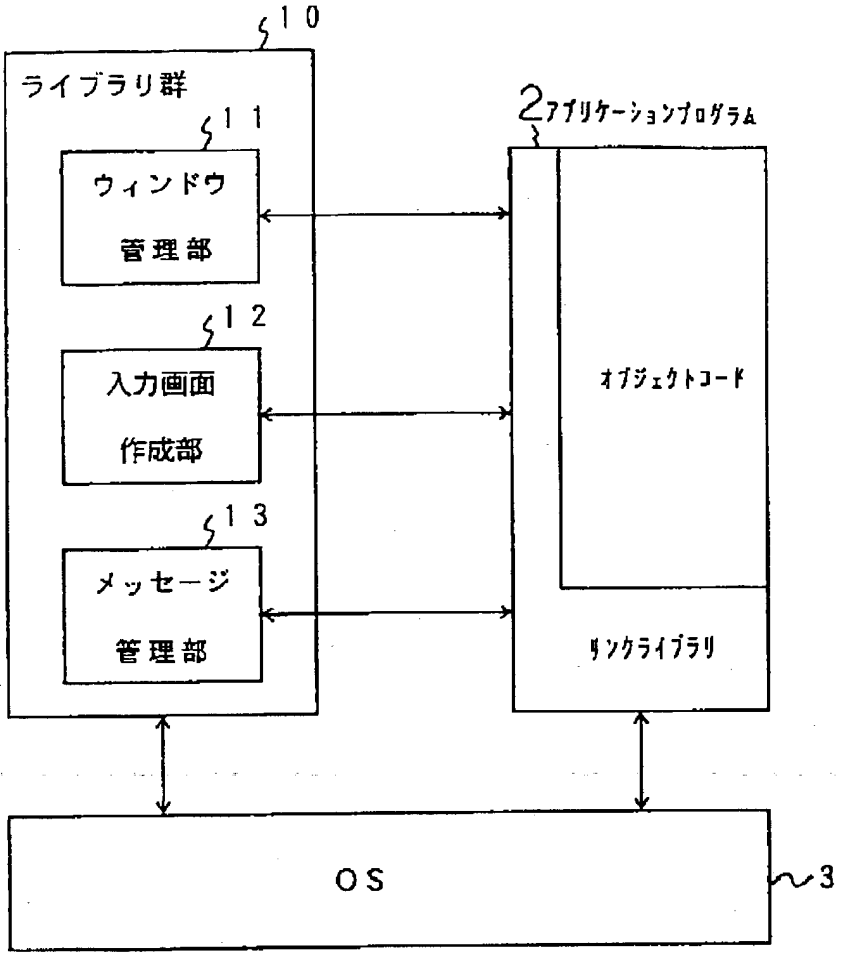
[Drawing 8]



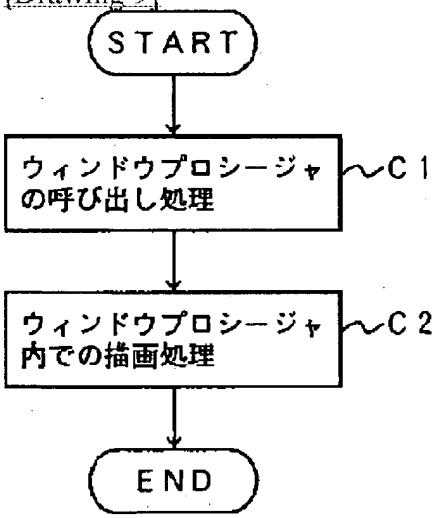
[Drawing 11]



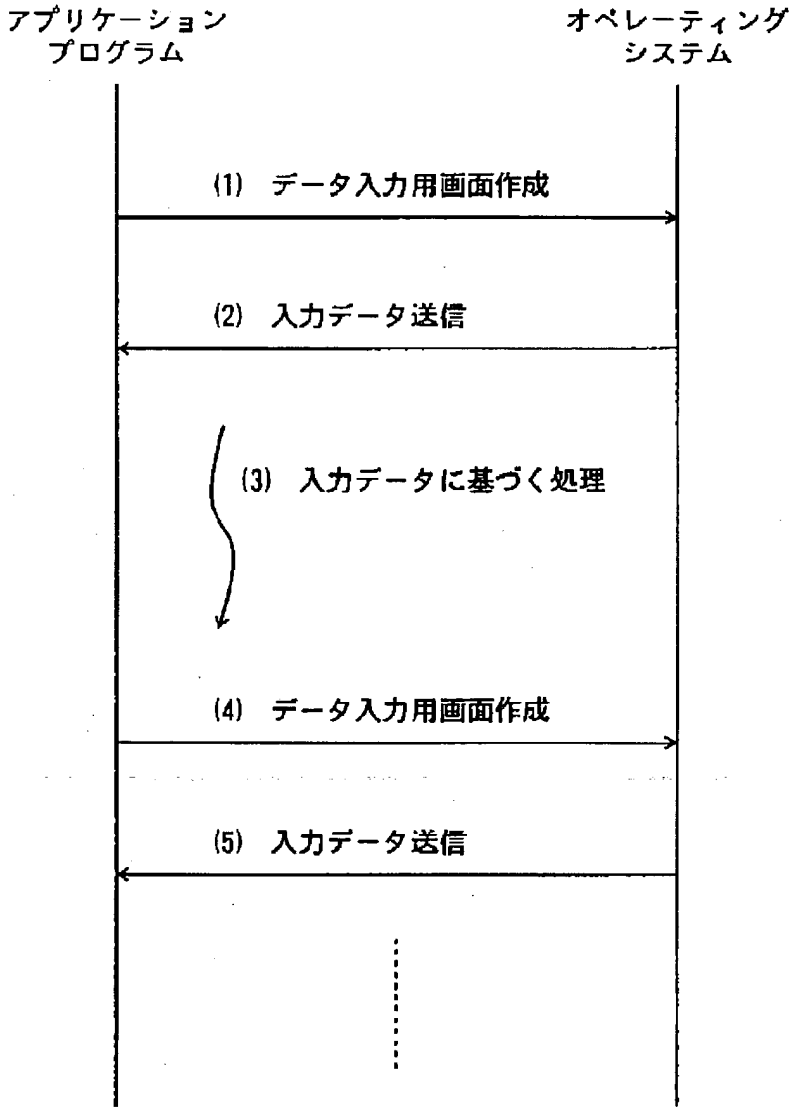
[Drawing 2]



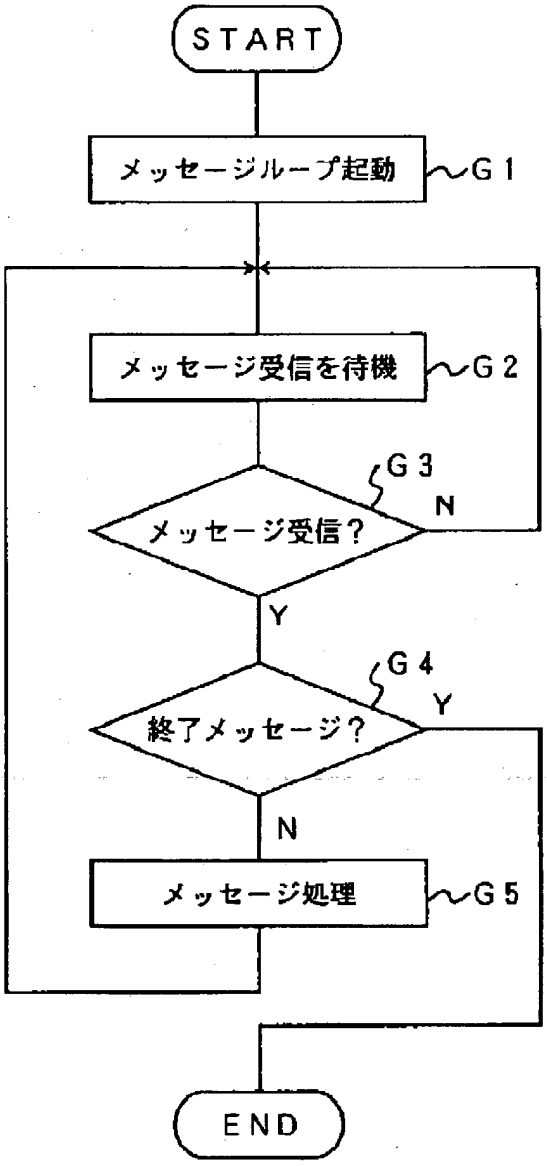
[Drawing 9]



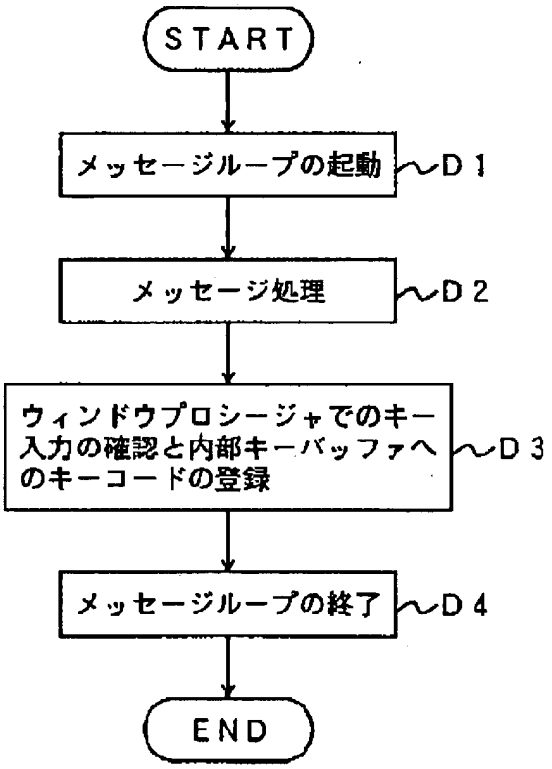
[Drawing 4]


アプリケーション
プログラム
オペレーティング
システム
(1) データ入力用画面作成
(2) 入力データ送信
(3) 入力データに基づく処理
(4) データ入力用画面作成
(5) 入力データ送信

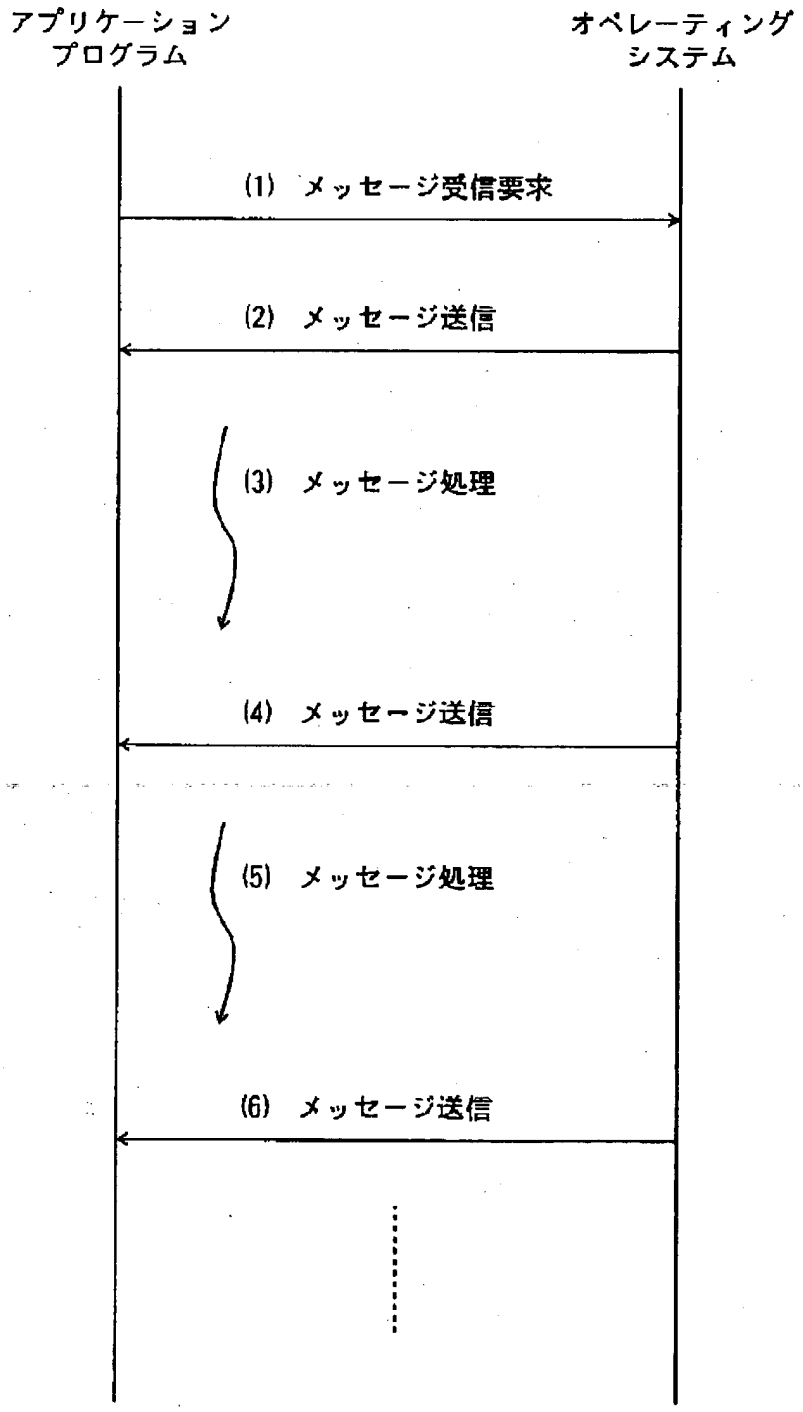
[Drawing 5]


START
メッセージループ起動
G 1
メッセージ受信を待機
G 2
G 3
メッセージ受信？
N
Y
G 4
終了メッセージ？
Y
N
メッセージ処理
G 5
END

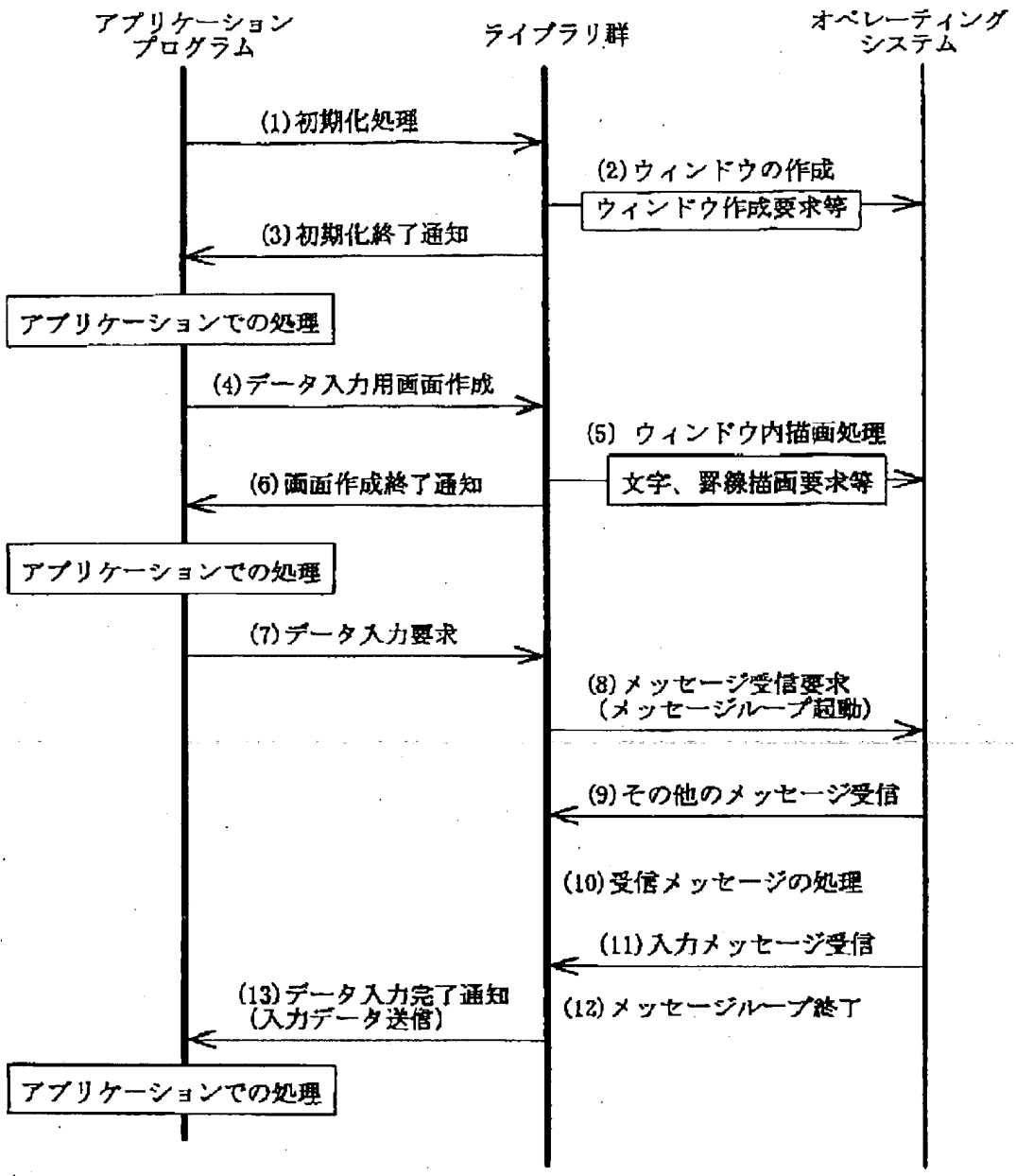
[Drawing 10]


START
メッセージループの起動
D 1
メッセージ処理
D 2
ウィンドウプロシージャでのキー入力の確認と内部キーバッファへのキーコードの登録
D 3
メッセージループの終了
D 4
END

[Drawing 6]


アプリケーション
プログラム
オペレーティング
システム
(1) メッセージ受信要求
(2) メッセージ送信
(3) メッセージ処理
(4) メッセージ送信
(5) メッセージ処理
(6) メッセージ送信

[Drawing 12]



[Translation done.]